Towards a Platform for Sharing Quantum Software

Frank Leymann, Johanna Barzen and Michael Falkenthal

Institute of Architecture of Application Systems
University of Stuttgart, Germany
{nachname}@iaas.uni-stuttgart.de

```
@inproceedings {INPROC-2019-24,
    author = {Frank Leymann and Johanna Barzen and Michael Falkenthal},
    title = {{Towards a Platform for Sharing Quantum Software}},
    booktitle = {Proceedings of the 13th Advanced Summer School on Service
Oriented Computing (2019)},
    publisher = {IBM Research Division},
    institution = {Universit{\"a}t Stuttgart, Fakult{\"a}t Informatik,
Elektrotechnik und Informationstechnik, Germany},
    series = {IBM Technical Report (RC25685)},
    pages = {70--74},
    type = {Konferenz-Beitrag},
    month = {September},
    year = {2019},
    keywords = {Quantum computing; software engineering; middleware;
platforms; cloud computing},
    language = {Englisch},
    cr-category = {D.2.0 Software Engineering General,
                   C.5.0 Computer System Implementation, General},
    department = {Universit{\"a}t Stuttgart, Institut f{\"u}r Architektur
von Anwendungssystemen},
    url = {http://www2.informatik.uni-stuttgart.de/cgi-
bin/NCSTRL/NCSTRL_view.pl?id=INPROC-2019-24&engl=}
 }
```

Towards a Platform for Sharing Quantum Software

Frank Leymann [0000-0002-9123-259X], Johanna Barzen [0000-0001-8397-7973]
and Michael Falkenthal [0000-0001-7802-1395]

Institute of Architecture of Application Systems, University Stuttgart, Germany
Lastname@informatik.uni-stuttgart.de

Abstract. Quantum computers solving real-world problems are expected to become general available within the next few years. But software for quantum computers require very different skills compared to creating software for traditional computers or networks. Thus, a community-driven approach to creating software for quantum computers will foster a wide-spread use of this innovative technology. Also, a platform for quantum software may provide a business model for several user groups.

Keywords: Quantum computing, software engineering, middleware, platforms, cloud computing.

1 Introduction

Quantum computing is becoming a reality: the first quantum computers are already commercially available or are about to be launched [5], [3], [4], [7]. The time at which quantum computers will solve problems that traditional computers can practically no longer solve (so-called "Quantum Supremacy" [10]) can be expected in the next few years [9]. Even earlier, non-ideal quantum computers can be used in practice [11].

Creating algorithms or software for quantum computers is significantly different from todays practice. Efforts in establishing a discipline of quantum software are significantly underrepresented [1]. Although there are a large number of algorithms for quantum computers (e.g. on websites like [6], in textbooks like [8], in scientific publications like [1]), which algorithm can be used in which situation requires a comprehensive understanding of the theory and technology, which users typically do not have. Even if a suitable algorithm is found, its conversion into an executable program requires deep knowledge of the environment of the respective quantum computer.

2 Quantum Software Platform

This is where the concept of the proposed Quantum Software Platform (QuSP) comes in. Components of the QuSP and user groups involved are shown in Fig. 1.

2.1 Overview

The algorithms for the QuSP come from many different sources (A) (NB: letters and numbers refer to labels in Fig. 1) such as the web, published articles or books. These algorithms are stored in a special database, the quantum algorithm catalog (1).

A public community (analogous to an open source community) or specialists of the platform operator (B) can access this special database and analyze, clean and unify the algorithms (2). As a result, each quality-assured algorithm is stored in the Quantum Algorithm Repository (3).

Based on the quality-assured algorithms, developers (C) can now implement these algorithms for execution on a quantum computer (4). These programs are also quality-assured (5) and stored in a quantum program repository (6).

End users (D) of the platform can now search for quality-assured algorithms and programs in the QuSP (7). If an algorithm for a certain problem is not found or if an algorithm is not implemented by a program, the end user can make corresponding requests (7) to the community. For delivery, an algorithm or program is packaged (9)

2.2 Population

Algorithms to be considered are captured by metadata. E.g. its reference is essential so that it can be retrieved and viewed. If it is available online, this reference is the corresponding link, otherwise the exact literature reference is given. It is also described which problems the algorithm is claimed to solve, initial information about the properties of the algorithm etc. is given.

Candidates for the quantum algorithm catalog can be identified in different ways. For example, community members can capture algorithms, or crawlers may automatically detect relevant publications.

2.3 Preparation

Next, the community agrees on the maturity of an algorithm [12]. If mature, it will be analyzed, unified and stored in the quantum algorithm repository. Analysis determines which problems the algorithm actually solves and with which properties: e.g. statements are made about the acceleration an algorithm achieves compared to certain classical algorithms [13], with how many qubits an algorithm is already useful [9], etc. During unification, the algorithm is brought into a common format (e.g., represented as a software pattern). If not mature, the algorithm is cleansed, i.e. its unsuitability is captured and the catalog is annotated accordingly so that this algorithm is not considered again.

These tasks can be performed by different user groups, e.g. a public community (analogous to an open source community), or specialists of the platform operator. For this purpose, the platform contains corresponding collaboration tools [12].

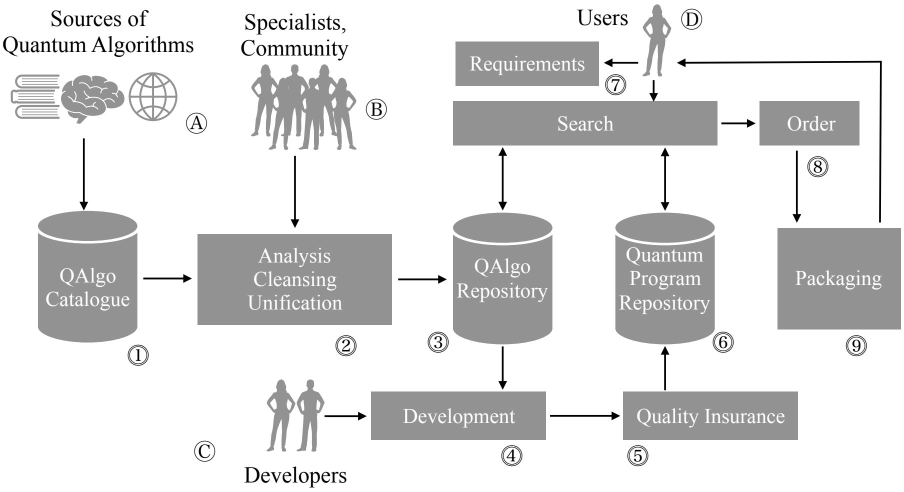

Figure 1 - Architecture of the quantum software platform QuSP.

2.4 Program Development

Quantum algorithms that meet specific requirements (such as the number of qubits required, fit for a hardware architecture) may be implemented so that they can run on the quantum computer of certain vendors. These quantum programs are examined for their quality (functional tests, etc.) before they are transferred to the quantum program repository. This can be done by an open source community. Also, specialists of the manufacturer of a certain quantum computer can implement those algorithms.

2.5 Usage

Customers of the QuSP can search for both, quantum algorithms and quantum programs. If an algorithm for a problem is not found, requirements to provide the appropriate algorithms can be imposed. If an algorithm is found but no corresponding implementation as a quantum program is found, an implementation can be requested. Providing algorithms and programs on request may be charged. If a quantum program is retrieved, the program is packaged so that it can be provisioned into its target environment.

3 Roles Involved

3.1 Platform Operators

The platform may be operated as a business, e.g. a fee can be charged for (successful) searches, for access to the unified representation of the algorithm etc. For programs that implement an algorithm, an even higher price can be charged. Satisfying requirements have to be paid.

3.2 Users

The search of different sources for suitable quantum algorithms as well as the assessment of their suitability is not only very time-consuming, but also requires highly specialized personnel. The use of the QuSP thus represents a significant advantage for users and often makes the use of quantum computers possible in the first place.

The same applies to the creation of a quantum program: the environment for developing such a program must be understood, and it is often even specific to a certain hardware of a supplier. Likewise, quantum algorithms are usually formulated independently of a specific hardware, so that adaptations to an appropriate target platform are necessary [14]. The purchase of a program that implements an algorithm for a specific quantum computer thus represents a considerable savings potential for a user.

3.3 Software Vendors

Quantum programs can be created by software companies and offered in the platform. Requirements that customers place on quantum programs and their target environments can thus be met by software companies. Software companies can specialize in different hardware platforms or development environments of quantum computers and thus achieve competitive advantages.

3.4 Hardware Vendors

Hardware vendors can also offer quantum programs: these programs are optimized for the hardware which becomes more attractive, of the programs must be paid.

3.5 Consulting Companies

Consulting companies are often specialized in industries (pharmaceuticals, finance...) or cross-sectional topics (optimization, simulation...) that can benefit from quantum computing. Personnel with knowledge in the field of quantum computing is rare, i.e. consulting firms can benefit in particular from the quality-assured algorithms of the QuSP in order to incorporate quantum technologies into their specialized consulting services. Users thus have access to corresponding consulting services.

4 Conclusion and Outlook

The sketched platform for quantum software will enable a much broader group of people and companies to take advantage of the benefits of quantum computing. Initial steps to create a prototype of such a platform is underway at our institute.

References

1. Ambainis, A., et. al.: “Quantum Software Manifesto“. (2017), http://www.qusoft.org/wp-content/uploads/2018/02/Quantum-Software-Manifesto.pdf
2. Coles, P.J., et al: „Quantum Algorithm Implementations for Beginners“. arXiv:1804.03719v1 (2018).
3. IBM: “Quantum Devices and Simulators”. (2018). https://www.research.ibm.com/ibm-q/technology/devices/#ibmq-20-tokyo
4. Intel: “Intel’s 49-Qubit quantum processor”. (2018). https://newsroom.intel.com/wp-content/uploads/sites/11/2018/05/49-qubit-processor-tangle-lake-infographic.jpg
5. Jelly, K.: “A preview of Bristlecone, Google’s new quantum processor”. Google (2018). https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html
6. Jordan, S.: "Quantum Algorithm Zoo“. (2018). https://math.nist.gov/quantum/zoo/
7. Linn, A.: “With new Microsoft breakthroughs, general purpose quantum computing moves closer to reality”. (2017). https://news.microsoft.com/features/new-microsoft-breakthroughs-general-purpose-quantum-computing-moves-closer-reality/
8. Lipton, R.J., Regan, K.W.: „Quantum Algorithms via Linear Algebra“. MIT Press 2014.
9. Mohseni, M., et al.: “Commercialize early quantum technologies“. Nature Vol. 543, March 2017.
10. Preskill, J., in „The Theory of the Quantum World“ (eds Gross, D., Henneaux, M. & Sevrin, A.) 63–80 (World Scientific, 2011).
11. Preskill, J.: “Quantum Computing in the NISQ era and beyond”. Quantum 2, 79 (2018).
12. Reiners, R.: „An Evolving Pattern Library for Collaborative Project Documentation“. Dissertation RWTH Aachen, 2013.
13. Rønnow, T. F., et al.: „Defining and detecting quantum speedup“. Science Vol. 345(6195) July 2014.
14. Tannu, S.S., Qureshi, M.K.: “Not All Qubits Are Created Equal - A Case for Variability-Aware Policies for NISQ-Era Quantum Computers”. arXiv:1805.10224 (2018).

(All links have been followed on June 2019).